TELKOMNIKA, Vol.13, No.3, September 2015, pp. 904~921
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v13i3.1496

Robust Path Construction for Reliable Data Transmissions in Node Disjoint Multipath Routing for Wireless Sensor Network

Abdulaleem Ali Almazroi*, MA Ngadi
Department of Computer sciences, Faculty of Computing, Universiti Technologi Malaysia
*Corresponding author, e-mail: maaaal2@live.utm.my, dr.asri@utm.my

Abstract

Wireless Sensor Networks (WSNs) are prone to node breakdowns due to energy constraints, which contribute to frequent topology changes. Moreover, since sensor nodes have restricted transmission range, multiple hops are needed by the node in order to forward the packets from one node to the other and this raises very challenging issues when designing routing protocols. Most of the proposed single path routing schemes use a periodic low-rate flooding of data in order to recover from path failures, which causes higher consumption in sensor node resources. So multipath routing is an optimal approach to enhance the network lifetime. In this paper, a robust path construction for a reliable data transmission in node-disjoint multipath routing (RNDMR) is proposed for WSNs. The proposed RNDMR has the ability to provide a low overhead path construction as well as provide data transmission reliability by using XOR-based coding algorithm, which entails low utilization of resources, such as low storage space and lesser computing power. In the proposed RNDMR, the procedure involves the splitting up of all transmitted messages into many different segments of equal size, before adding the XOR-based error correction codes and distributing it among multiple paths simultaneously in order to boost reliable data transmission and to be assured that the essential fragment of the packet arrives at the sink node without any additional consumption of energy and undue delay. By using simulations, the performance of RNDMR was assessed and compares it with ReInForm routing. The results illustrate that RNDMR attains low energy consumption, records low average delay and routing overhead, as well as increased packet delivery ratio when compared with ReInForm Routing.

Keywords: *wireless sensor network, node-disjoint multipath routing, robust path construction, forward error correction, reliability*

1. Introduction

The rapid progression of wireless sensor networks (WSNs) in recent times have led to substantial research interest, largely because of different varieties of applications which may not need human interventions, especially in the fields of military missions and accessing hostile environments. But for some reasons, lots of interests are given to routing protocols because of the different types that exist, such as the architecture of the network and the application. Sensors are distinguished in WSNs by means of its constrained energy, memory resources, processing, unpredictable wireless links and higher degree of mobility. The appearance of these characteristics in sensor networks contributes to different challenges for every part of the protocol stack, particularly the network layer, for instance, assurance of end-to-end packet delivery [1, 2]. There have been several innovative routing protocols designed specifically for WSNs in recent years. These include an energy-aware routing protocol to prolong the life of the network [3, 4]; tolerance to faults in sensors because of quicker exhaustion of battery or erratic wireless links [5]; and Quality of Service (QoS) schemes to stabilize the consumption of energy and certain predetermined QoS metrics that are needed by the various kinds of applications [6, 7]. The most existing research works in routing protocols have used single path routing to transmit data to the destination. Most researchers adopt this method because it very simple to design as well as providing less energy consumption. Unfortunately, for single path routing, most of the paths are susceptible to either node or link breakdowns. Therefore, acknowledgments and retransmissions mechanism are implemented in order to recover data lost, which contributes to huge amount of extra traffic and delays in the network. The processes

Received February 1, 2015; Revised May 19, 2015; Accepted June 1, 2015

also entail the discovery of new paths required every time for sustaining transmission of data from the source to sink and so this procedure of path discovery contributes to additional message overhead and energy cost. Therefore, in order to raise the reliability of data transmission, multipath routing protocols are normally used and this event happens whether erasure code is used or not [8-10].

To the best of the authors' knowledge, no such mutlipath routing exist in wireless sensor network that addresses the the problems of message overhead and energy consumption which emanate from the path construction phase and data duplication transmission, respectively. Therefore, in this paper we propose a robust path construction for data transmissions in node-disjoint multiple paths routing (RNDMR) for WSNs that aims to provied low ovherad path construction as well as provide data transmission reliability by using XOR-based coding algorithm. The proposed RNDMR comprises of five different features, namely, path construction, drastic decrease in routing overhead, filtering overlapped paths, attaining multiple node-disjoint routing paths, and reliable data transfer across multiple paths. RNDMR employs XOR-based coding algorithm, which does not involve high storage space or high computation power. RNDMR breaks up the transmitted messages into several fragments of equal sizes, adds XOR-based error correction codes and spreads it across multiple paths concurrently in order to enhance the reliability of transmission and guarantee that a fundamental segment of the packet reaches the sink node without introducing any extra energy consumption and delays during the period of data retransmission.

The remainder of the paper is organized as follows. Section 2 provides a brief overview of related works. The proposed RNDMR routing scheme is described in Section 3. The simulation setup and discussion of simulation results are presented in Section 4 and Section 5, respectively. Finally, the conclusion of this work is provided in Section 6.

2. Related Work

The need for fault tolerance emanates from the need to ensure that the system is always accessible for use without any form of interruption of service as a result any type of fault. Therefore, fault tolerance raises the reliability, accessibility and consequently dependability of the system. In fault tolerance, the most famous approach is multipath routing, where a pair of multiple paths among the source nodes and the sink are determined at the expense of increased consumption of energy and generation of traffic. Another good feature for multipath routing is the provision of load balancing and bandwidth aggregation. The categorization of routing protocols being recommended for WSNs are divided into three groups, which is subjected to the procedures used for discovering the path. In the case of proactive routing, every path is calculated and conserved in advance and stored in the routing table, for reactive routing, every path is created on demand basis, and in hybrid routing, which is a mix of both proactive and reactive routing [11].

There are two techniques [9] used to create multiple paths. In the case of disjoint multipath, it builds a number of alternate paths that can either be node or link disjoint with the primary path. Thus, when breakdown occurs in either a node or link on the primary path, the alternative paths are used directly. The creation of these alternative paths implies that there would be an increased energy consumption as compared to that of the primary path, which results from their extensive latency. Additionally, global topology knowledge is required to facilitate the creation of the multiple disjoint paths. Using this multipath technique in a network with k node-disjoint routes from the source toward destination can usually tolerate a minimum of $k-1$ intermediate network component breakdowns [12]. The second technique is braided multipath, which involves creating an alternative path for every node in the primary path. It implies that the alternate paths in a braid partially overlay with the primary path. However, the construction of the alternative paths are not costly in comparison with the primary path, in terms of latency and overhead. Though in a situation where the whole or most parts of the nodes along the primary paths encounter any failure, the discovery of new path becomes necessary, which contributes to extra overheads [13]. The classification of reliability mechanisms in multipath routing are divided into replication and retransmission types.

2.1. Retransmission Based Schemes

It is the most popular mechanism used for retransmitting data packets to the sink on

each multiple paths by taking into account the least hop count or the least consumption of energy subjected to the conditions of the network. The procedure involves the sink node sending acknowledgement back to the source anytime a data packet is successfully received by sink during transmission. In a situation where the acknowledgement was not delivered to the sender and a timeout occurs, retransmission of the data packet would have to be done. Moreover, the rate of packet loss through the wireless link in WSNs is much greater in comparison with other networks, so the most famous mechanism used is the link level retransmissions. But the negative effect when using this mechanism is the rise in network traffic, which requires more consumption of resources. By sending an acknowledgement message also might raise delivery delays leading to several loss of packets because of the likelihood of collisions occurring. Another consideration is the issue of memory; more memory space is required by the sensor nodes to ensure buffering of the packets till the acknowledgement from the destination is received [14, 15]. In the subsequent paragraphs, we will explain the routing protocols based on retransmission mechanism and present the main concepts of the protocols.

The most innovative and well known routing protocol proposed in WSNs is Directed Diffusion routing protocol [16]. Many other routing protocols are normally based on the concepts of Directed Diffusion or follow the same concept. The basic procedure for this protocol is to ensure that the sink broadcasts an interest packet which has to be periodically refreshed inside the network. This packet is a query, which includes the information that the sink requests from the sensor nodes. Upon receiving the query packet, the entire node in the network caches the packet in their respective memories before attempting to flood it to surrounding neighbors to ensure that all nodes received it. Every single node creates a gradient, which includes the data rate as well as the direction where the data will be transmitted. When a node senses an event in the monitoring area, it will compare it with the information being maintained in its cache. Whenever a match is discovered, the node is considered to be the source node, which periodically broadcasts a message at low rate in order to forward packets. When the sink receives numerous detection events, meaning that there are multiple paths existing at the source at a given instant, it will broadcast a reinforcement on one of these paths (normally a path having the least delay) by increasing the data rate of the query packet. In a situation when failure occurs in the reinforced path, the sink cannot detect any kind of data. It reinitiates the reinforcement message to use other paths in order to reroute the lost data. Therefore, to ensure provision for quicker recovery arising out of path failure, the sink should periodically broadcast reinforcement messages to enable faster discovery of alternative path, which should be constructed on-demand basis. As this protocol is based on query driven data delivery, it does not work well in environmental monitoring applications, which need reliable data delivery to the sink. Also, due to its energy requirement in broadcasting lower rate of messages, the protocol is not regarded as energy efficient protocol. The sensors might contribute additional overhead when matching data and queries.

Another protocol based on Directed Diffusion routing is the highly-resilient, Energy Efficient Multipath Routing Protocol for WSNs [17]. The researchers illustrate how a multipath routing technique discovers partial disjoint paths; the discovered paths are not disjoint paths as in Directed Diffusion, but they are known as braided multipath. This protocol maintains the cost lower by maintaining the multipath as well as recovering from path failures more quickly. The protocol also avoids the periodic flooding that is utilized in Directed Diffusion routing. In this protocol, the multipath among the source node and the sink is constructed by the network, then a path is chosen as the primary path to route the data packet, and alternative paths are kept alive by constantly transmitting a “keep-alive” data among the paths. When the primary path fails, the nodes recover quickly through reinforcements of other paths to reroute the data packets that have been lost. Furthermore, the energy consumption in this protocol results from all the paths (that is, from source to sink), which are set in advance and stored though periodically transmitting a low data rate data, known as “keep-alive”. This process is regarded to be more robust, however it increases the network cost in terms of consumption of energy.

Energy consumption is the main criteria for Reliable Energy Aware Routing (REAR) in WSNs [18]. It recommends an energy reservation scheme to be used for routing data to the sink. Additionally, to ensure network reliability, a backup path is established for every primary path from the source to sink. The main concept of the protocol is to ensure that if the sink receives an interest through a source node, which does not belong to its routing table, it will creates two disjoint paths to the source node. A specific path is then chosen and employed to

transmit the data packet. Another path (the second path) is also employed to be the backup path in the event of failure happening. Additionally, part of the energy will be conserved for both paths in all the intermediate nodes along these paths. In the event of a path failure happening, the intermediate node transmits the data packet back to the source node and the sink receives reporting error. The outcome involves both the source node and the sink have to delete the failure path from their routing table. The energy conserved for that specific path is released from every node along that path. Lastly, all the traffic is re-directed to the backup path. But when the primary path (that is, service path) is established again, all traffic is quickly re-directed again onto it.

2.2. Replication Based Schemes

In WSNs, redundancy in packet delivery is considered to be another mechanism utilized to provide reliable multiple paths routing. Routing protocols usually implement a replication mechanism to ensure delivery of original packet to sink, which is by the transmission of several copies of original packet across several paths. The main setback when using this technique is increased overhead, normally due to the packet being sent by every node until it arrives at the sink. Thus, in WSNs, the provision of reliability and load balancing can also be attained through another technique known as coding scheme [19, 20].

Coding schemes split the data packets and transmit them through diverse discovered paths. This scheme conserves transmission and receiving energy, however it requires additional computation at every node and path maintenance. The successful implementation of the coding technique is also based on the amount of paths discovered and number of splits received at the destination node. When the delivered number of data packets at the destination is less than the expected (necessary) number, then the original data cannot be recovered. Also, robustness and compression efficiency of the coding technique are the main issues of concern. Thus, trade-offs exists among reliability and energy efficiency [21]. In the following paragraphs, we describe the routing protocols based on replication mechanism and highlight their key ideas.

ReInForM [22] has been proposed in multipath routing sensor networks. This routing protocol employs a method, known as packet duplication technique, to supply desired data transmission reliability for every application. The process involves an effort to enhance reliability of data transmission by employing the packet duplication method for the entire sensor nodes involved during the process of data transmission without considering packet segmentation method. The eminent reliability is attained at higher cost of consumption of energy and usage of bandwidth, which is in contrast with the main demands of resource limited sensor nodes.

H-SPREAD [23] recognizes a multipath extension flooding stage, where nodes from diverse branches swap the found paths. Consequently, it finds more disjoint paths at the cost of extra messages exchange, by breaking the property of utilizing "one message per node''. When a sensor node finds a new path, it notifies its neighbor about it. Frequently, this information is disseminated over the network to attain more disjoint paths in each node. Obviously, this extension burdens sensors with extensive energy utilization because of the exchange of the extra messages.

Delay-Constrained High-Throughput Protocol for Multipath Transmission (DCHT) [24] is the enhanced version of Directed Diffusion (DD), which advocates the idea of employing multipath routing. The routing function in the protocol begins through flooding an interest message to the entire network between the sink to source node concurrently. At any time, a source node have the capability to deliver the data being demanded through the sink node. It transmits data packets that have been discovered in the direction of the sink node, in the initial phase by using recognized gradients. Whenever several copies of the occurrences are established, reinforcement messages are transmitted by the sink to a specific neighbor desiring greater frequency from a particular neighbor. On every occasion, it receives notification of detection events. In case of node breakdown in the reinforced path, reinforcement is carried out by sink because of its ability to sense an absence of detection events. The main problem of this protocol, with respect to energy consumption and message overhead, is that reinforcement messages will have occasionally transmitted by the sink through many paths toward the source node.

In WSNs, to improve the reliability of packet delivery, Reed Solomon algorithm [25], with Multipath on Demand Routing (MDR), is utilized to code or decode and route the data packet from source to the sink. In this protocol (that is, MDR), the main concept entails when the

source has data to transmit to the sink, where it initiates the procedures of routing request phase through flooding in the network. This flooding mechanism comprises the IDs of the source and the sink, respectively and also the request ID. At any time, the sink receives any route request messages, it responds and resends a route reply message with additional field that signifies the total number of hops. So every node receiving the route reply message can now increase the hop count and relay the message to the nearest neighbor. These procedures continue up till certain time, the source node will gathers every route reply message received. It maintains the IDs of its neighbors and also the length of the path. In the last procedure, the source node fragments the data packet based on the number of paths, path lengths and the maximum probability of failure.

3. Description of RNDMR Routing

In this section, we propose a robust path construction for data transmissions in node-disjoint multiple paths routing for WSNs, which is useful to find node-disjoint multiple paths between the source and the destination with low overhead as well as attain data transmission reliability by utilized XOR-based coding algorithm.

3.1. Cost Function and Route Selection

The main concept for route selection is the capability of having to choose an optimal path to extend the lifetime of the network. This concept is founded on path cost function. The key purpose of path cost function is to provide additional weight/cost to the node that has a lesser amount of energy in attempting to extend its lifetime. Let $P_j = \{P_1, P_2, P_3, ..., P_n\}$ represent the paths with low overhead emanating through the source *S* to destination *D* by various intermediate nodes $\{n_1, n_2, n_3, ..., n_k\}$ then:

$$P_1 = S - n_1 - n_2 - n_3 - D$$
$$P_2 = S - n_4 - n_5 - n_6 - D$$
$$P_3 = S - n_7 - n_8 - n_9 - D$$

In attempting to choose the optimal path, the path cost function takes into account the total cost of all intermediate nodes on every path, as represented in Equation (1):

$$C(P) = \sum_{i=1}^{k} f_i(c_i) \quad (1)$$

Where $f_i(c_i)$ indicates the total cost of all nodes on path by taking into consideration the residual energy and the sum of the hops whereas $C(P)$ denoting the path cost of the path *P*. Consequently, to choose the optimal path between a pair of paths that have founded on Equation (1) during route discovery with low overhead, the optimal path meant to be the minimum total cost is designated as Equation (2):

$$Optimal\_path = \min(C(P)) \forall P \in P_j \quad (2)$$

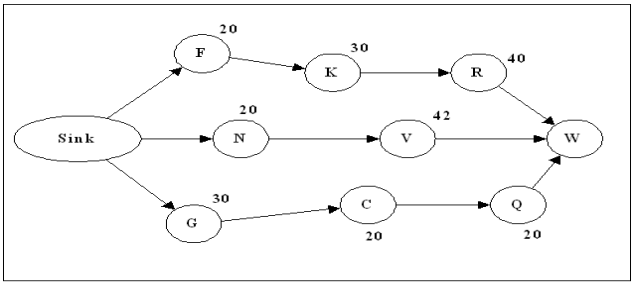


Figure 1. Network with 10 nodes

As shown in the Figure 1 above, there are three paths from sink node to destination node (W). As per Equation (1), P1, P2 and P3 denote the sum of costs and therefore the cost of these paths are; $C(P_1) = 20 + 42 = 62$, $C(P_2) = 30 + 20 + 20 = 70$, and $C(P_3) = 20 + 30 + 40 = 90$ Hence, from Equation (2), the optimal path for Figure 1 is P_1.

3.2. Path Construction

Any time a sink node requires a specific information about the network, it verifies its route table to confirm approval if there is an appropriate route. When true, it transmits the interest packet to the best subsequent hop in the direction towards the destination. Nevertheless, when sink valid route to the destination is absent and not available, it can begin a route discovery process. In order to start such a process, the sink node constructs a RREQ (Route Request) message that comprises of explanations for all the information needed by the user. The packet holds ptype, SrcID address, dnode ID address, brID, hcunt, Enr_cost and Route_list fields as shown in Figure 2. The field, ptype, denotes the packet type, which is route request message. The field, hcunt, indicates hop count and it is increased by one for every node through which packet passes. The brID field signifies broadcast ID and it always increases each time the sink node begins a RREQ. In view of this process, a distinct identifier for RREQ is created by the broadcast ID and the address of the node, which initiated the RREQ message. The Enr_cost field represents the total energy cost. Once a packet passes through a node, its energy cost is added to this field (i.e., Enr_cost). Initially, by default, this field contains zero value. The Route_list field is a path construction list of the route path. To ascertain node-disjoint multiple paths having lesser broadcast overhead, the least energy cost is usually associated with difficult procedures especially if there are limited knowledge on the topology of the network and variations are frequent. Therefore, the significant purpose of RNDMR is the construction of node-disjoint multipath that consists of a low routing overhead and the minimum energy path in the period of a route discovery to ensure much greater realization delivery ratio. To accomplish this purpose, the node holding the necessary information should have knowledge of the entire routing path list of existing routes to allow for the selection of the most suitable node-disjoint route paths within the existing candidate paths. When the RREQ messages are created or relayed through the sink inside the network as described in Figure 2, every node attaches its own address and adds its energy cost to the routing request message.

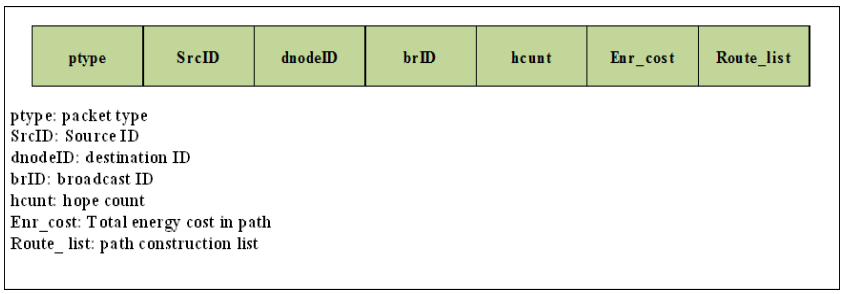


Figure 2. RREQ message

As shown in the Figure 2 above, when a RREQ message reaches its node having the required information (destination node), then it is responsible for deciding if the routing path is a node-disjoint path or otherwise. Once a node-disjoint path have been verified, the node produces a Route Reply message, as described in Figure 3 that holds the node list of the entire route path and then unicasts in the direction where the originated RREQ message is emanating from. Once an intermediate node receives a RREP message, it updates the entries in the routing table using the nodes list of the entire route path contained in the Route Reply message.

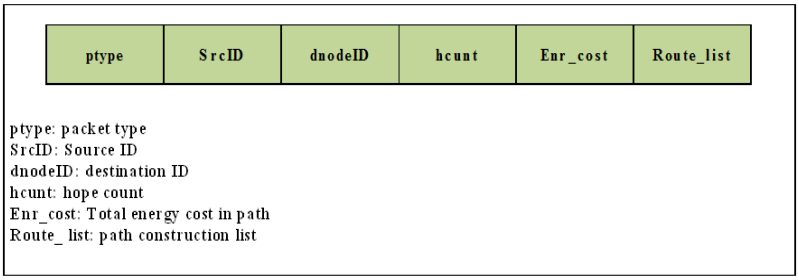


Figure 3. RREP message

Figure 4 describes an example of path construction and it consists of four nodes, namely, Sink, N, V and W. If the sink node intends to transmit a specific task to node W, and if the Sink node route to W is non-existence in its routing table, it transmits a RREQ. If the RREQ is received by node N, it must attach its address, increase the field of hop count by one, and update the energy cost field of the RREQ before forwarding the request, because route W is inaccessible and has no route to it. Likewise, if node V receives the RREQ, it attaches its address and its energy cost and increases the field of hop count by one in the RREQ message. Once the request arrives at the target W, node W examines the path construction list (Sink-N-V) through the RREQ and reviews if the routing path is a node-disjoint path or otherwise. Once true, node W produces a Route Reply message, which holds the path construction list for the entire path and forwards it towards the direction of the sink node, which initiated the RREQ message. If not true, node W rejects the received RREQ message.

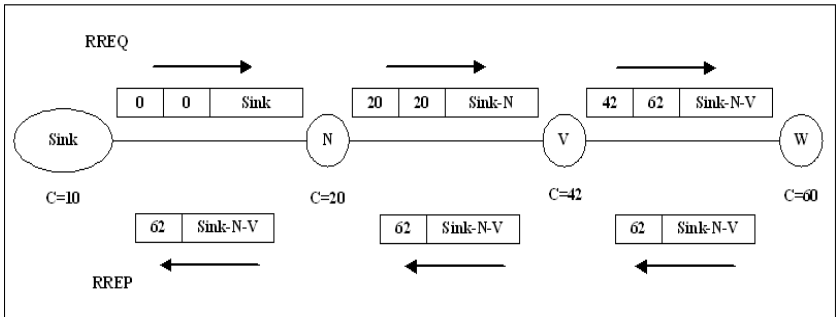


Figure 4. Path construction process

3.3. Route Request Broadcast with Low Overhead

When a node gets a RREQ message, which happens to be its first one, it examines the path construction list from the message packet, calculates total hops starting from the source to itself and then records the total residual energy in its route table. When the node gets the RREQ identical message for another time, the node calculates the number of hops from the source to itself and contrasts it to the number of the shortest hops, which is stored in the entry of the routing table. If the new message has smaller number of hops, the node attaches its addresses, in addition to its energy cost, to the route path list for the RREQ message and then transmits the RREQ message to surrounding nodes. Otherwise, the new RREQ message is rejected. The pair (Source address, Broadcast ID) is utilized to differentiate the message packets. Consequently, for this method, several identical RREQ messages will be rejected when new RREQ has greater number of hops when compared to the record.

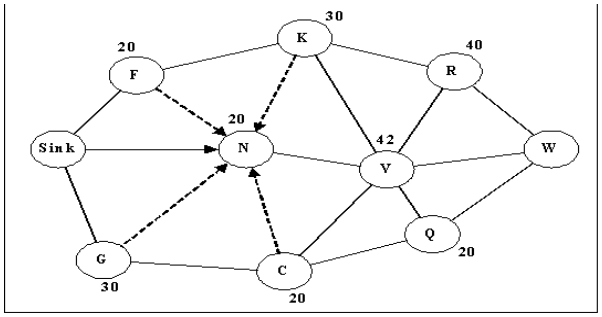


Figure 5. RREQ Propagation with the Shortest Hops

From the illustration in Figure 5, starting from the sink node to node N, there are a combination of five route paths, namely: Sink-N, Sink-F-N, Sink-G-N, Sink-F-K-N, and Sink-G-C-N. The total number of the hops comprises of 1, 2, 2, 3 and 3, in that order, and the energy costs involved are 0, 20, 30, 50 and 50, respectively. If node N gets the RREQ packet during the initial period from Sink-N, it registers 0 to be the energy cost and 1 to indicate the smallest total of hops. Once node N gets a RREQ identical message through the remaining four route paths, it computes the total of hops and contrasts it to the smallest number of hops in its routing table. Since the total number of hops of the route list for all the four route paths exceeds 1, it will reject the four Route Request identical messages.

3.4. Paths Overlapped Filtering

Additional method during Router Request message was to filter overlapped routing paths to minimize the routing overhead. From Figure 6, node R gets three RREQs and that denotes three routing paths Sink-F-K, Sink-N-K and S-N-V. These all have an equal number of hops, that is 3 hops. However, the method does not only forward them, but also examines if the routing paths are node-disjoint or otherwise before forwarding. When there are overlapped routing paths, decisions are taken to reject one of them for node-disjoint. Therefore, node R compares RREQs when it received the messages within a specific period. As a result, the routing path, Sink-N-K-R, which includes a common overlapped link K-R when equated with that of path Sink-F-K-R. It also has additional overlapped link, Sink-N, when compared with the path, Sink-N-V-R. Consequently, it rejects the routing path, Sink-N-K-R, having extra two common overlapped links when compared with the other paths. Therefore, the concept was to just re-broadcast two RREQ messages containing the routing paths information of Sink-F-K-R and Sink-N-V-R. Thus, node R can be assured of two node-disjoint routing paths. Likewise, node Q transmits two RREQ messages, once assured of two node-disjoint paths. Every neighbor node adapts this method with low routing overhead to transmit RREQ messages. Algorithm 1 highlights the process of RREQ message with low overhead in intermediate nodes.

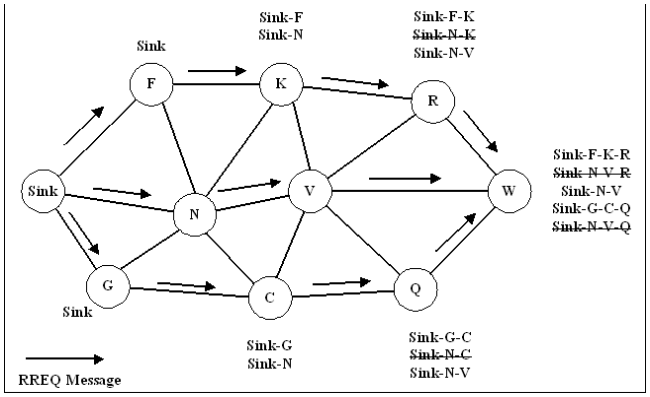


Figure 6. RREQ Propagation with no Overlapped

Algorithm 1: Algorithm to process the RREQ with Low broadcast routing overhead

Set myaddress: address intermediate node, ID: broadest id of RREQ, mycost: energy cost of intermediate node

Set SA: Source Address, DA: Destination Address, h: hop count, RT: Routing Table, Enr cost: energy in path

Step1: Receive RREQ // Check for node address equal to target //

if (*myaddress == RREQ[DA]*) **then**

Act as destination to send reply;

end if

Step 2: // if node address exists in RREQ of path then drop RREQ //

if (*myaddress exit in RREQ[path]*) **then**

Drop RREQ; go to step 6.

end if

Step 3: //Compare the pair (Source Address, ID) of RREQ with each entry of Route table (RT)//

if (*RREQ[SA, ID] not exist in RT [SA, ID]*) **then**

//Record the partial information RREQ into RT by creating new entry

RT[SA]=RREQ[SA]; RT[ID]=RREQ[ID];

RT[DA]=RREQ[DA]; RT[hop]=RREQ[hop];

RT[Enr cost]=RREQ[Enr cost];

//Assign the hop to L1 //

L1= RREQ[hop];

//Update the fields of RREQ by adding node cost to cost field, appending node address to Route_list field , increasing hop //

RREQ [Enr cost]= RREQ[Enr cost]+ mycost;

RREQ[Route list]=RREQ[Route list]+ myaddress;

RREQ [hop]= RREQ[hop]+1;

Broadcast the RREQ to another intermediate node

else

Step 4: if match is found, then currently received RREQ becomes new duplicate RREQ say

DRREQ, Assign its hop to L2.//

if (*RREQ[SA, ID] exist in RT [SA, ID]*) **then**

L2= DRREQ [hop];

Step 5: // compare the currently received RREQ (New duplicate) with previous RREQ //

if (*L2 >= L1*) **then**

Drop DRREQ; go to step 6.

else

RT[SA]=DRREQ[SA]; RT[ID]=DRREQ[ID];

RT[DA]=DRREQ[DA]; RT[hop]=DRREQ[hop];

RT[Enr cost]=DRREQ[Enr cost];

RREQ [Enr cost]= RREQ[Enr cost]+ mycost;

RREQ[Route list]=RREQ[Route list]+ myaddress;

RREQ [hop]= RREQ[hop]+1;

//Assign its hop to L1 //

L1= RREQ[hop];

Broadcast the DRREQ to another intermediate node

Step 6: perform step1 to step5.

end if

end if

end if

3.5. Selecting Node-Disjoint Paths

Within the set of rules for choosing node-disjoint paths, the destination node is responsible for choosing node-disjoint route paths. To ensure reduction in overhead of the routing table for every node, the total number of node-disjoint routing paths are restricted to the three paths that have the smallest energy cost path and less number of hops even though more

than three node-disjoint paths are explored. Therefore if the RREQ messages are received by the sensor node having the required information, it caches the node IDs list for the whole route paths in its routing table and transmits three RREP messages that comprise the route paths in the direction to the sink, which initially sent the RREQ messages. In this instance, the foremost route is the route with the smallest energy cost and number of hops. When the destination node receives another router request message, it contrasts the entire route path in the RREQ message to all the existing node-disjoint route paths in its routing table. Therefore, if there is no shared node among the route path in the RREQ message and any node-disjoint route path, which is cached in the destination's routing table, then the route path ensures the requirement and conditions of the node-disjoint is recorded in the destination's routing table. Else, the route path is discarded. In addition, the third node-disjoint path that has smallest energy cost and number of hops is chosen. At the end of this procedure, on-demand data transmissions will take place when sensor nodes detect an interesting event. Algorithm 2 highlights the process of creating node-disjoint paths.

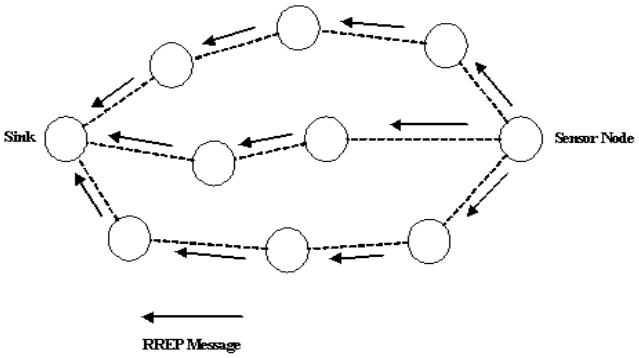


Figure 7. Construction of node-disjoint paths

Algorithm 2: Algorithm for creating node-disjoint paths

Let M is a set of m-1 multiple paths from excluding primary

Let p_1, p_2, p_3,...,p_{m-1} be the m-1 multiple paths that are stored at two dimensional array M.

Let P_p is primary path stored 1-D array N.

Let D=set of paths that are node-disjoint to primary. Initialize D= ø.

min C(P):minimum cost of the path.

// D is computed as follows //

for (k = 1 to m-1) **do**

Select $p_1$from M and Check it is minimum cost of the path and it is node disjoint to primary.

if (p_1== *min C(P))* **then**

if ($p_1 \cap P_p$ == ø) **then**

add p_1 to D,

M = M - p_1;

else

Drop p_1;

end if

end if

end for

3.6. Data Packet Segmentation and Encoding

The data packet is divided into N sub-packets having the same sizes $(D_0, D_1, D_2, \ldots D_{N-1})$, and an overhead part of *H+1*, where *H<N*. Moreover, it also appends the

error correction codes $(C_0,C_1,C_2,...C_H)$ of equal size as the data fragment, which are added to the original packet as revealed in Figure 8. The error correction codes and data fragments have the same length of l bytes and should be multiple of 8. The error correction codes are computed as a function of the information bits to supply redundant information by using an XOR-based Forward Error Correction (FEC) technique. XOR-based FEC technique does not involve high storage space or high computation power. The correction codes are calculated as below:

$$C_0 = D_0 \oplus D_1 \oplus D_2 \oplus \ldots \oplus D_{N-1}$$
$$C_1 = D_1 \oplus D_2 \oplus D_3 \oplus \ldots \oplus D_H$$
$$C_2 = D_2 \oplus D_3 \oplus D_4 \oplus \ldots \oplus D_{(H+1) \bmod N}$$
$$C_3 = D_3 \oplus D_4 \oplus D_5 \oplus \ldots \oplus D_{(H+2) \bmod N}$$
$$\vdots$$
$$C_H = D_H \oplus D_1 \oplus D_2 \oplus \ldots \oplus D_{(H+k) \bmod N}$$

Where $\oplus$ represents exclusive XOR operation.

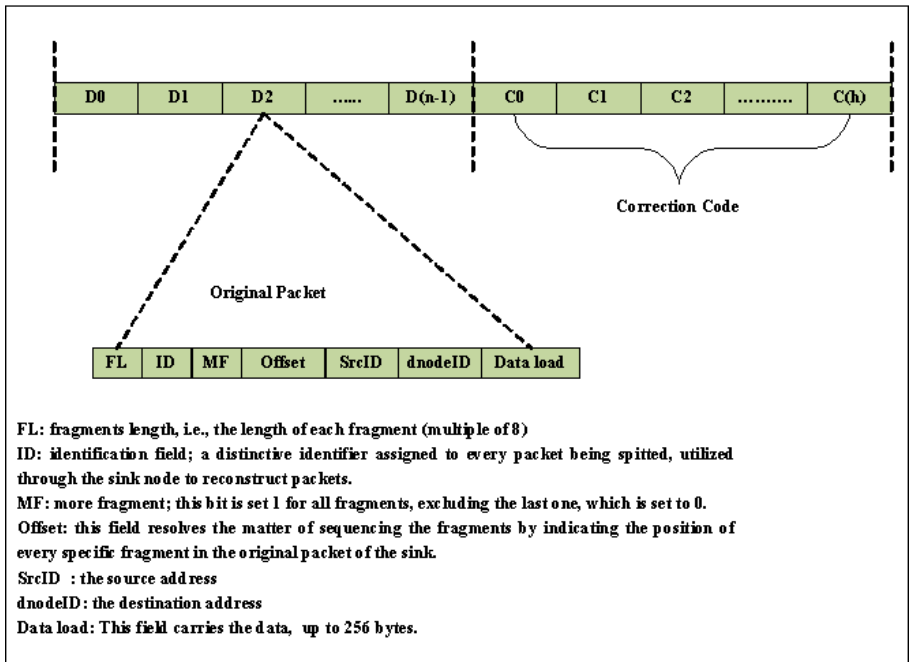


Figure 8. Packet structure

3.7. Data Decoding and Recovery

After calculating the XOR error correction codes, the fragments of the original packet $(D_1,D_2,D_3,....D_N)$, alongside with the error correction codes $(C_0,C_1,C_2,C_3,...C_H)$, are forwarded through the available multipath. [26] demonstrated that if H or less fragments are lost out of the N+H total data and overhead correction codes, the original N packet fragments can be reassemble by XOR operation. The original packet could be reassembled as below:

$$D_1 = C_0 \oplus C_2$$
$$D_2 = C_0 \oplus C_3$$
$$\vdots$$
$$D_H = C_0 \oplus C_1$$

Figure 8 shows the packet fragmentation and the fields in every sub-packet. As indicated in the legend to Figure 8, the FL field is fixed fragment length and the fragment should have a length, which is a multiple of 8. The ID, signifying the identification field, is a distinctive identifier allocated to every packet being broken up. The field is utilized via the sink node to reconstruct and reassemble packets without accidentally mixing fragments from different packets. More fragments field (MF) has its bit set to 1 for all fragments, excluding the last one, which is set to 0. When the fragment with a value of zero in the MF field bit is recognized, the sink node knows that it has received the last fragment of the packet. The segment-offset field resolves the problem of sequencing fragments by indicating the location of every fragment in the sink node as in the original packet.

4. Simulation Setup

RNDMR is implemented in ns-2 network simulator, which is a discrete event-driven and object oriented simulation platform. The simulation was conducted on a square area of 500 m × 500 m, in which the wireless nodes are dispersed randomly. There is one sink node, having no power constraints (that is, sink node is deployed at the center of the area), and one source node in the network. Moreover, the simulation is for different sizes of sensor networks ranging from 50, 100, 150, 200 and 250 in varied traffic rate conditions. The rate of packet transmission is 1, 2 and 3 packets per second. The simulation lasted for 500 sec. Every node has a fixed transmission range of 250 meters. The data packet size is 512 bytes. Every node is assigned the same initial energy value of 10 Joules at the beginning of the simulation to maintain the simulation time within a reasonable time. The simulation further introduced an Omni antenna (Antenna Type) to every node and adopt the IEEE 802.11 MAC layer protocol provided in the ns-2 and the energy consumptions for transmission and reception are E_{elec}= 50 nJ/bit. Table 1 gives the standard simulation parameters.

Table 1. Parameter values used in simulation for proposed RNDMR

Parameters	Value
Simulation dissemination	500m ×500m
Node placement Node	Random Distribution
Node numbers	50,100,150,200,250
Number of Sinks/Number of Sources	1/1
Packet size (data + overhead)	Up to 510 bytes
Sub-packet size	170 bytes
Node initial energy	10 J
Transmission range	250 m
Traffic Type	CBR
Packet rate	1,2,3 packets/s
Buffer Threshold	64 bytes
Transmit Power	E_{elec}= 50 nJ/bit
Receiving Power	E_{elec}= 50 nJ/bit
Transmitter Amplifier (ε_{amp})	150 pJ/bit/ m^2
Simulation time	500s

5. Metrics and Results Analysis

a) The packet delivery ratio (PDR): it is the ratio of the number of delivered data packet to the destination and the total number of packets transmitted from source. This clarifies the level of delivered data to the destination.

b) Average Energy Consumption: it is computed in the entire topology. This metric is the measure for the network lifetime. It evaluates the average variation among the energy initial level and the energy final level that is left in every node. Let E_i be the energy initial level of a node i, E_f be the energy final level of a node i and N denote the number of nodes in the simulation. Therefore,

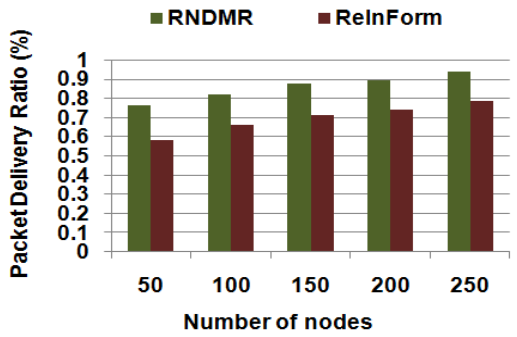
$$E_n(i) = \frac{\sum_{i=0}^{n} E_{consp} = e_{i,i-}e_{f,i}}{N} \quad (3)$$

c) Average End-to-End Delay: It is given as the average time among the moment a data packet is transmitted through a source node $t_{source,i}$ and the moment the sink received the data packet $t_{sink,i}$. If N is the number of successful received packets, then the average end-to-end delay, D is given as:

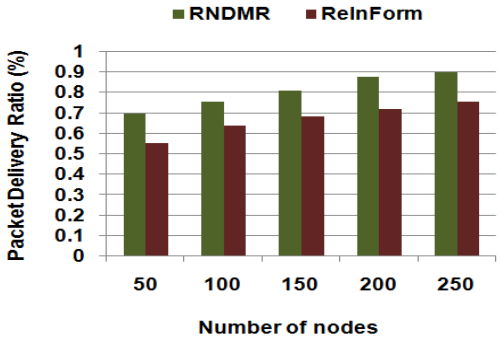
$$D = \frac{\sum_{i=0}^{n} t_{sink(i)-}t_{source(i)}}{N} \quad (4)$$

d) Control overhead: It is identified as the overall number of routing control packets which is processed by node normalized of the total number of received data packets by the sink node.

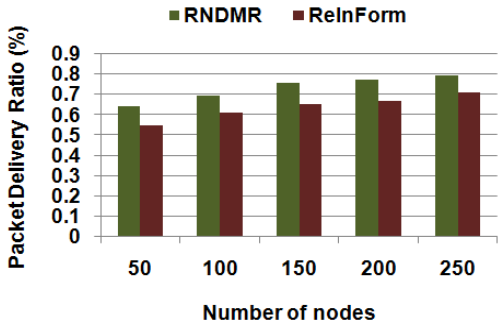
5.1. Simulation Results in PDR



(a) Packet Delivery Ratio data rate 1 (pkt/sec)



(b) Packet Delivery Ratio data rate 2 (pkt/sec)

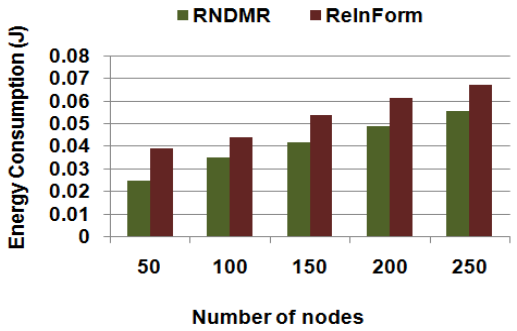


(c) Packet Delivery Ratio data rate 3 (pkt/sec)

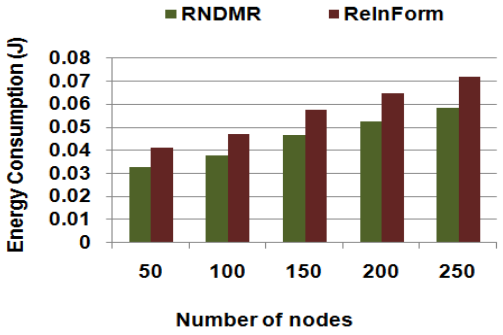
Figure 9. Simulation results of packet delivery ratio for RNDMR and ReInForm routing at different traffic rates and number of nodes

The PDR is considered to be a very important metric because it shows the loose rate which in turn affects the highest throughput of the network. The simulation results of PDR for RNDMR and ReInForm routing at different traffic rates is given in Figure 9. It shows that the RNDMR has the best performance delivery ratio when compared with ReInForM. Figure 9(a) illustrates that the proposed RNDMR is 23 % higher than ReInForM when network size is 250 nodes at 1 packet per second. Also, the figures show that the delivery ratio is increased when the network size also increases. This is because RNDMR has multiple paths with node-disjoint. Additionally, RNDMR constructs paths with low overhead and it considers the minimum number of hops and minimum energy of path when selecting these paths whereas, ReInForM does not consider the node energy during path selection, which causes many path failure during data forwarding and hence the number of packet drops also increases. In addition, RNDMR used an error correction code, which enhances the delivery ratio in the case of path breakdowns. RNDMR simply reconstructs the data by using the generated XOR codes when there is path breakdown, which is not used in ReInForm, therefore no data retransmission is needed. Figure 9(b) shows that the proposed RNDMR is 20 % higher than ReInForM when network size is 250 nodes at 2 packets per second. Finally, Figure 9(c) also illustrates that the proposed RNDMR is 14 % higher than ReInForM when network size 250 nodes and 3 packets per second.

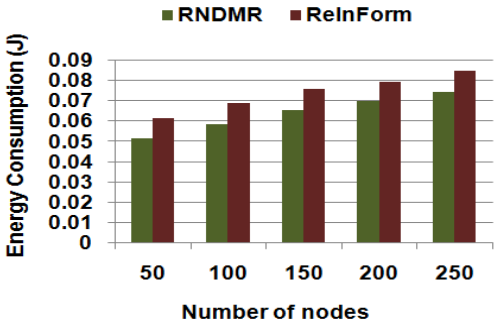
5.2. Simulation Results in Average Energy Consumption



a) Energy Consumption data rate 1 (pkt/sec)



(b) Energy Consumption data rate 2 (pkt/sec)



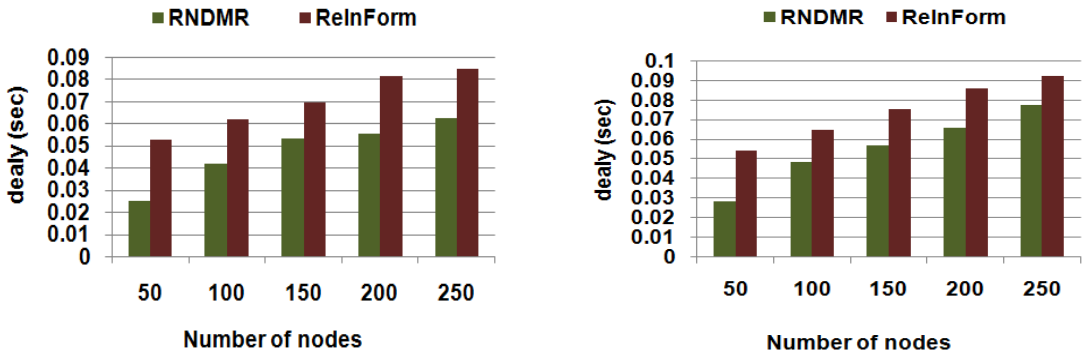
(c) Energy Consumption data rate 3 (pkt/sec)

Figure 10. Simulation results of Average Energy Consumption for RNDMR and ReInForm Routing at different traffic rates and number of nodes

The average energy consumption is computed in the entire topology. It evaluates the average variation among the energy initial level and the energy remaining level that is left in every node. Energy consumption of the node is the subsequent metric to be conducted. Figure 10 shows that in comparison with the ReInForM scheme, the RNDMR have the best energy consumption performance. In Figure 10(a), 22 % enhancement is recorded by RNDMR, which has lesser average energy consumption than ReInForm with the number of nodes being 250 and a data rate of 1 packet per second. This is because RNDMR simply reconstructs the data

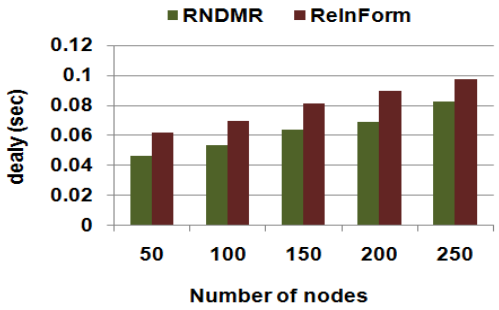
by using forward error correction codes when there is path breakdown without involving data retransmissions. Moreover, RNDMR decreases significantly the amount of control packets (i.e., routing overhead) by employing the smallest number of hops and using filter overlapped routing paths notions during the paths construction period. Therefore, it avoids undue energy consumption, which comes from the data retransmission and huge control packets (i.e., high overhead). But, ReInForm uses packet duplication technique by sending multiple copies of every data packet through multipath in the network to attain the reliable transmission, which is generating more energy. In Figure 10(b), RNDMR records 18 % enhancement, which represents lesser average energy consumption than ReInForm with number of nodes at 250 and a data rate of 2 packets per second. In Figure 10(c), a 13 % enhancement of RNDMR is recorded, which amounts to lesser average energy consumption than ReInForm with the number of nodes at 250 and a data rate of 3 packets per second. Therefore, RNDMR can save network energy and reduce energy consumption by using the advantages of multipath routing and forward error correction coding.

5.3. Simulation Results in Average Packet Delay



(a) End to End delay data rate 1 (pkt/sec)

(b) End to End delay data rate 2 (pkt/sec)



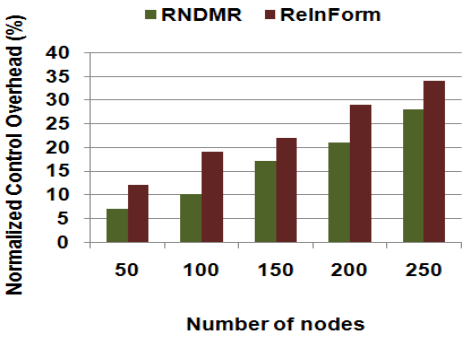
(c) End to End delay data rate 3 (pkt/sec)

Figure 11. Simulation results of average delay for RNDMR and ReInForm at different traffic rates and number of nodes

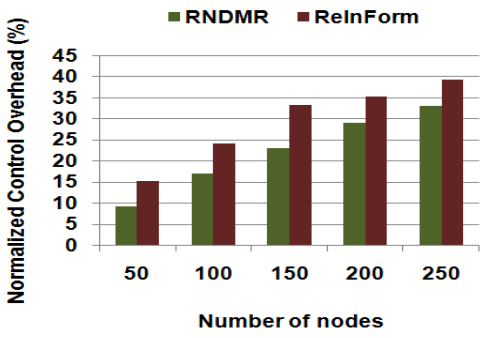
The average end-to-end delay defines the average time taken by a data packet to arrive at the destination. Figure 11 presents the general trend of whole curves, implying an increase in delay with the increase of data rate traffic. The reason is mostly that the average delay increases with increasing traffic data rate. Figure 11 illustrates that in comparison with another scheme, the RNDMR having the shortest delay than ReInForm. In Figure 11(a), 33 % is enhancement of RNDMR having lesser average end-to-end delay than ReInForm with the number of nodes being 250 and a data rate of 1 packet per second. This is because RNDMR selects the path based on minimum number of hops besides minimum energy in path during paths construction. Therefore, the lesser the number of hops, the less time the data spends to reach to the destinations. Moreover, in RNDMR, the forwarding of sub-packets through

multipath concurrently have avoided the congestion in the network, which is opposite of ReInForm routing, which forwards the multiple copy of every data packet through multipath routing without considering splitting of every data packet. Correspondingly, in Figure 11(b), RNDMR records 25 % enhancement, which denotes lesser average end-to-end delay than ReInForm with number of nodes at 250 and a data rate of 2 packets per second. In Figure 11(c), RNDMR again records 21 % enhancement, which represents lesser average end-to-end delay than ReInForm with the number of nodes at 250 and a data rate of 3 packets per second.

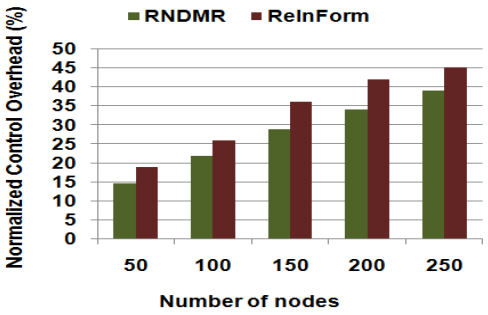
5.4. Simulation Results in Control overhead



(a) Control overhead data rate 1 (pkt/sec)



(b) Control overhead data rate 2 (pkt/sec)



(c) Control overhead data rate 3 pkt/sec)

Figure 12. Simulation results of control overhead for RNDMR and ReInForM Routing at different traffic rates and number of nodes

The control overhead is the ratio of the average amount of control message treated by the node and the amount of data packets received by the sinks. There are several routing schemes, with each having a distinct technique and if a network is much bigger, it is necessary to request for a greater amount of exchange control messages so as to be able to find and create more routes. This implies that a greater amount of energy might be consumed and needed during the initial construction phase. Furthermore, when the network is a bigger one, it means there will be an extensive separation distance between the sink and the source nodes. There are many intermediate nodes which have to be navigated to enable a data packet to arrive at the destinations sink node. Control message overhead of several routing are illustrated in Figure 12. This value is obtained through computation of two elements, that is, the ratio of the average amount of control message processed by the node and the amount of data packets received by the sinks. Here, the control message overhead portrays ReInForm as having to expend greater amount of energy during transmission and receiving control messages when compared with RNDMR. In Figure 12(a), RNDMR provides a 28 % enhancement than ReInForm with RNDMR have smaller control message overhead with the number of nodes at 250 and a data rate of 1 packet per second. The reason being that RNDMR constructs node-disjoint multipath with a low routing overhead and it considers the minimum number of hops and minimum energy of path when selecting these paths. Moreover, RNDMR is able to filter

overlapped routing paths and it reduces the routing overhead significantly. On the other hand, ReInForm does not consider the node energy during path selection, which causes many path failures, therefore it needs to generate more new path discovery process resulting in more routing control overhead. In Figure 12(b), a 24 % enhancement for RNDMR is recorded, which has a lesser control message overhead than ReInForm with number of nodes at 250 and a data rate of 2 packets per second. In Figure 12(c), RNDMR records 17 % enhancement, indicating a lesser control message overhead than ReInForm with number of nodes at 250 and a data rate of 3 packets per second.

6. Conclusion

In this paper, we have presented a robust path construction for reliable data transmissions in node disjoint multipath routing (RNDMR) for WSNs. RNDMR routing includes five fundamental components that are used in discovering node-disjoint multipath with low overhead and therefore achieve reliable transmission. These are, path construction, decrease route request messages, filtering overlapped paths, selecting node-disjoint paths and reliable data transfer across multiple paths by utilizing XOR-based Forward Error Correction. Through computer simulations, we have evaluated and studied the performance of RNDMR routing and compared it with ReInForM. Simulation results show that RNDMR routing achieves more energy savings, higher delivery ratio, lower average delay and lower routing overhead than ReInForM.

Acknowledgements

Abdulaleem Ali Almazroi would like to thank Dr. MA Ngadi for his support in his study and also would like to thank Northern Borders University in Saudi Arabia for their support in his scholarship.

References

[1] Almazroi AA, Ngadi MA. Energy Efficient Node Disjoint Multipath Routing to Improve Wireless Sensor Network Lifetime. *Journal of Theoretical and Applied Information Technology*. 2015; 71(2): 215-226.
[2] Mahyastuty VW, Pramudita AA. Low Energy Adaptive Clustering Hierarchy Routing Protocol for Wireless Sensor Network. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2014; 12(4): 963-968.
[3] Tufail A, Qamar A, Khan AM, Baig WA, Kim K-H. WEAMR — A Weighted Energy Aware Multipath Reliable Routing Mechanism for Hotline-Based WSNs. *Sensors*. 2013; 13(5): 6295-6318.
[4] Kominami D, Sugano M, Murata M, Hatauchi T. Robust and Resilient Data Collection Protocols for Multihop Wireless Sensor Networks. *IEICE transactions on communications*. 2012; 95-B(9): 2740-2750.
[5] Chanak P, Samanta T, Banerjee I. Fault-Tolerant Multipath Routing Scheme for Energy Efficient Wireless Sensor Networks. *International Journal of Wireless & Mobile Networks*. 2013; 5(2): 33-45.
[6] Mazinani SM, Naderi A, Setoodefar M, Shirazi Z. *An Energy-Efficient Real-Time Routing Protocol for Differentiated Data.* Proceedings of the 2012 IEEE 17th International Conference on Engineering of Complex Computer Systems. 2012: 302-307.
[7] Hariyawan MY, Gunawan A, Putra EH. Wireless Sensor Network for Forest Fire Detection. *TELKOMNIKA*. 2013; 11(3): 563-574.
[8] Alwan H, Agarwal A. A Multipath Routing Approach for Secure and Reliable Data Delivery in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*. 2013; 2013: 1-10.
[9] Sha K, Gehlot J, Greve R. Multipath Routing Techniques in Wireless Sensor Networks: A Survey. *Wireless Personal Communications*. 2013; 70(2): 807-829.
[10] Uppal RS, Kumar S, Singh H. Reliable and Energy Saving Multipath Routing in Multisink Wireless Sensor Networks. *Global Journal of Computer Science and Technology Network, Web & Security*. 2013; 13(13): 25-34.
[11] Pantazis NA, Nikolidakis SA, Vergados DD. Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*. 2013; 15(2): 551-591.
[12] Park J, Jo M, Seong D, Yoo J. Disjointed Multipath Routing for Real-Time Data in Wireless Multimedia Sensor Networks. *International Journal of Distributed Sensor Networks*. 2014; 2014: 1-8.
[13] Venkateswarlu MK, Sekaran KC, Kandasamy A. Node-Link Disjoint Multipath Routing Protocols for Wireless Sensor Networks – A Survey and Conceptual Modeling. In: Thilagam PS, Roshan Pais A, Chandrasekaran K, Krishan V. *Editors*. Advanced Computing, Networking and Security. Berlin: Springer-Verlag; 2012: 405-414.

[14] Bhattacharya R, Ray C. Wireless Sensor Networks – A Study of Fault Detection and Recovery Based on OSI Layers. *International Journal of Conceptions on Computing and Information Technology*. 2013; 1(1): 9-14.
[15] Mahmood MA, Seah WKG, Welch I. Reliability in Wireless Sensor Networks: A Survey and Challenges Ahead. *Computer Networks*. 2015; 79: 166-187.
[16] Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: *A Scalable and robust Communication Paradigm for Sensor Networks*. Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. New York, NY. 2000: 56-67.
[17] Chang JH, Tassiulas L. Maximum Lifetime Routing in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking (TON)*. 2004; 12(4): 609-619.
[18] Hassanein H, Luo J. *Reliable Energy Aware Routing in Wireless Sensor Networks.* Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems (DSSNS'06). Columbia, MD. 2006: 54-64.
[19] Liu X, Gong X, Zheng Y. Reliable Cooperative Communications Based on Random Network Coding in Multi-Hop Relay WSNs. *IEEE Sensors Journal*. 2014; 14(8): 2514-2523.
[20] Al-Hamadi HH. Dynamic Redundancy Management of Multisource Multipath Routing Integrated with Voting-based Intrusion Detection in Wireless Sensor Networks. PhD Thesis. Falls Church, Virginia: Virginia Polytechnic Institute and State University; 2014.
[21] Wang L, Yang Y, Zhao W. Network Coding-Based Multipath Routing for Energy Efficiency in Wireless Sensor Networks. *EURASIP Journal on Wireless Communications and Networking*. 2012; 2012(1): 115.
[22] Deb B, Bhatnagar S, Nath B. *ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks.* Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03). Bonn/Konigswinter, Germany. 2003: 406-415.
[23] Lou W, Kwon Y. H-SPREAD: A Hybrid Multipath Scheme for Secure and Reliable Data Collection in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*. 2006; 55(4): 1320-1330.
[24] Li S, Neelisetti R, Liu C, Lim A. *Delay-Constrained High Throughput Protocol for Multi-Path Transmission Over Wireless Multimedia Sensor Networks*. Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks. Newport Beach, CA. 2008: 1-8.
[25] Wu J, Dulman S, Havinga P, Nieberg T. *Multipath routing with erasure coding for wireless sensor networks.* Proceedings of the SAFE and ProRISC. 2004: 181-188.
[26] Avci SN, Ayanoglu E. *Link failure recovery in large arbitrary networks via network coding*. Proceedings of the 2014 Information Theory and Applications Workshop (ITA). San Diego, CA. 2014: 1-10.